САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА МОДЕЛИРОВАНИЯ ЭКОНОМИЧЕСКИХ СИСТЕМ

Хмара Егор Борисович

Выпускная квалификационная работа бакалавра

Совершенствование метода MSER для выявления особых точек и построения дескрипторов их окрестностей на цифровых изображениях

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Ковшов А. М.

Санкт-Петербург

2017

Оглавление

Оглавление 2

Введение 3

Постановка задачи 5

Обзор литературы 6

Глава 1. Теория 8

1.1. Метод Максимально Стабильных Экстремальных Областей 8

1.2. Формальное определение MSER 11

1.3. Свойства MSER 13

Глава 2. Реализация 15

2.1. MSER на Java 15

2.2. MSER на C++ 20

Глава 3. Тестирование 21

3.1. Результаты работы MSER метода на Java. 21

3.2. Результаты работы MSER метода на C++. 23

3.3. Сравнение результатов по количеству найденных регионов. 25

3.4. Сравнение результатов по времени работы алгоритмов. 27

Выводы 28

Список Литературы 29

Введение

В последние годы быстрыми темпами развиваются технологии, связанные с созданием искусственных систем. Значительную область в этой сфере занимают системы, получающие и обрабатывающие информацию с изображений. Компьютеры относительно недавно начали управлять обработкой большого количества данных, поэтому область компьютерного зрения можно считать молодой, и универсальных методов, решающих задачи компьютерного зрения ещё не создано. Однако в этой области существует много методов, которые направлены на решение более узкоспециализированных задач.

Типичными задачами компьютерного зрения являются задачи: распознавания, движения, восстановления изображения и восстановления сцены. В этой работе в основном будет затронута такая проблема распознавания, как детектирование.

На некотором этапе обработки возникает задача выявления определённых областей или точек изображения, которые имеют важное значение для дальнейшей обработки. Решением этой задачи занимается такая система компьютерного зрения как Детектирование.

С помощью детектирования на изображении выявляются особые точки, которые обладают высокой инвариантностью и стабильностью, и в дальнейшем могут быть использованы для сопоставления изображений. А это является ключевым этапом к решению задач распознавания объектов, панорамирования, слежения [1], восстановления сцены, поиска изображений по содержанию, обнаружения, стерео сопоставление.

Нахождение соответствий между двух изображений сцены, снятых с различных точек обзора, разными камерами и в разных условиях освещения, является сложным, но крайне важным шагом в направлении полностью

автоматической реконструкции трехмерных сцен. Важнейшей проблемой является выбор элементов, соответствие которых ищется. В подавляющем большинстве случаев локальные деформации изображения не могут быть аппроксимированы движением, то есть параллельными переносами, поворотами, различными симметриями и их комбинациями. Поэтому требуется полное аффинное преобразование, включающее в себя растяжение и сжатие изображения. Таким образом выявление соответствий изображений не может быть выполнено путём сравнения фиксированных форм, например, прямоугольников или окружностей [2].

На большинстве изображений есть области, которые могут быть обнаружены с высокой повторяемостью [3], поскольку они обладают некоторыми отличительными, инвариантными и стабильными свойствами. Такие особые области или DR (distinguished regions), могут служить элементами, по которым можно выявлять соответствия при сопоставлении изображений или трёхмерных реконструкциях [2].

Одним из методов обнаружения этих локальных особенностей на изображение является метод «Максимально Стабильных Экстремальных Областей» или MSER метод (Maximally stable extremal regions). Метод MSER устойчив к изменениям масштаба, вращениям и освещения [4].

Как таковой, метод использовался в приложениях распознавания объектов [5], поиска изображений [6], распознавания и сопоставления [7], отслеживания [1]. До недавнего времени, он использовался для обнаружения текста [8][9].

Постановка задачи

Целью данной работы является совершенствование метода MSER по скорости, либо функциональным возможностям относительно существующих реализаций. Например, реализации метода, представленном в открытой библиотеке алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов opencv. Проверить работоспособность и эффективность реализации. Сравнить результаты работы усовершенствованного метода с работой метода библиотеки opencv.

Реализацию метода MSER я планирую выполнить на языке Java версии Java SE 8. Для сравнения реализацию программы с использованием библиотеки opencv напишу на на C++ стандарта 2011 года (C++11), так как используя оболочку для opencv под Java (Javacv), возможно снижение скорости выполнения программы, что может повлиять на объективность исследования.

Обзор литературы

Метод MSER официально был представлен в 2002 году, на конференции по компьютерному зрению «British Machine Vision» (BMVC). Доклад представлял профессор чешского технического университета Иржи Матас, совместно с О. Хумом, М. Урбаном и Т. Паидлом [2]. В последующие годы над методом продолжали работать, появлялись новые модификации и расширения, проводились исследования о его возможностях и применениях.

В 2005 году вышла статья Кристиана Миколайджека с соавторами в которой они сравнили аффинные детекторы областей, в том числе и детектор MSER [3]. На основании этой статьи можно выделить следующие особенности метода MSER:

- MSER обнаруживает множество небольших областей, и превосходит пять другие рассматриваемые детекторы в плане обнаружения при изменении точки обзора.
- MSER стоит на втором месте по обнаружению при изменении масштаба и вращению в плоскости.
- Однако MSER оказался наиболее чувствительным к размытию.
- MSER показал самую большую повторяемость при изменении света.

MSER доказал что является надежным областным детектором и показал наивысшие результаты среди остальных детекторов.

В 2006 году И. Матас и С. Обдржалек рассмотрели метод MSER и его модификации для распознавания изображений [5].

В 2007 году профессором Линчёпингского университета Эриком Форшеном был представлен детектор основанный на цветовой аффинной ковариации областей, который является расширением MSER алгоритма [7].

В 2011-2013 годах было опубликовано несколько статей в области обнаружения и распознавания текста с использованием метода MSER [8][9][10].

В 2011 году Аарон Чавез и Дэвид Густавсон рассмотрели недостатки стандартных MSER алгоритмов в отношении изменений фона, и предложили новый дескриптор на основе MSER, а так же сравнили результаты со стандартными MSER алгоритмами и его расширениями основанными на цвете [11].

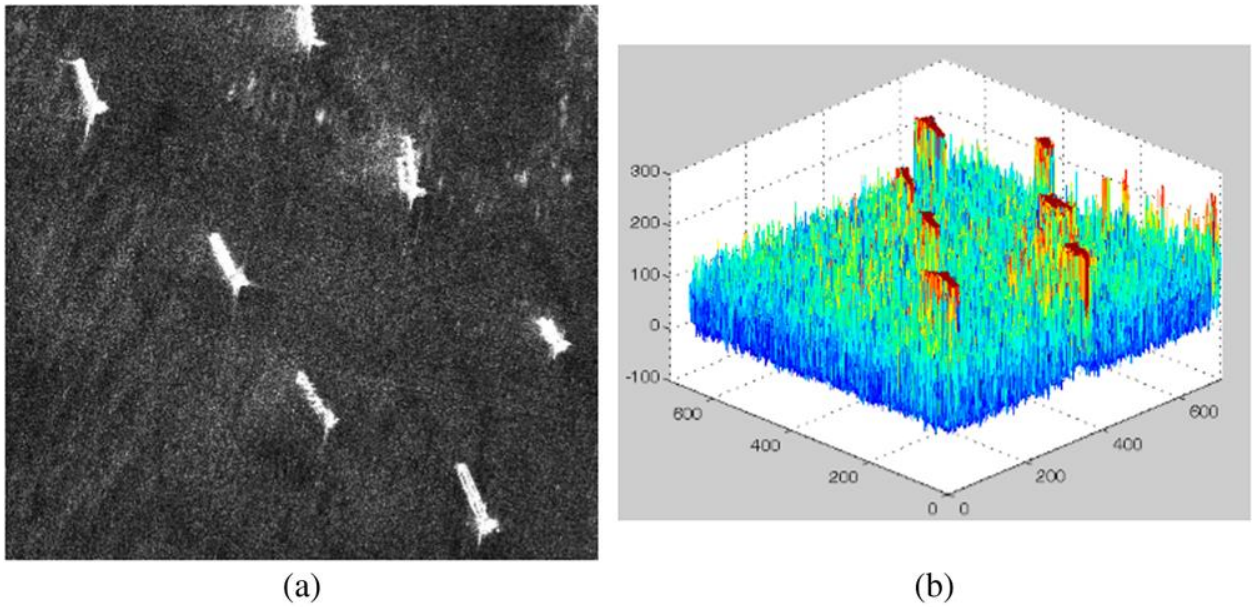
Алгоритм MSER актуален по сей день, находятся новые области его применения и продолжаются исследования [4][13].

Глава 1. Теория

1.1. Метод Максимально Стабильных Экстремальных Областей

В первую очередь определим значение Максимально Стабильных Экстремальных Областей (MSER). MSER – тип элементов изображения пригодных для нахождения соответствий. Элементы определяются экстремальным свойством функции интенсивности в области и на ее внешней границе.

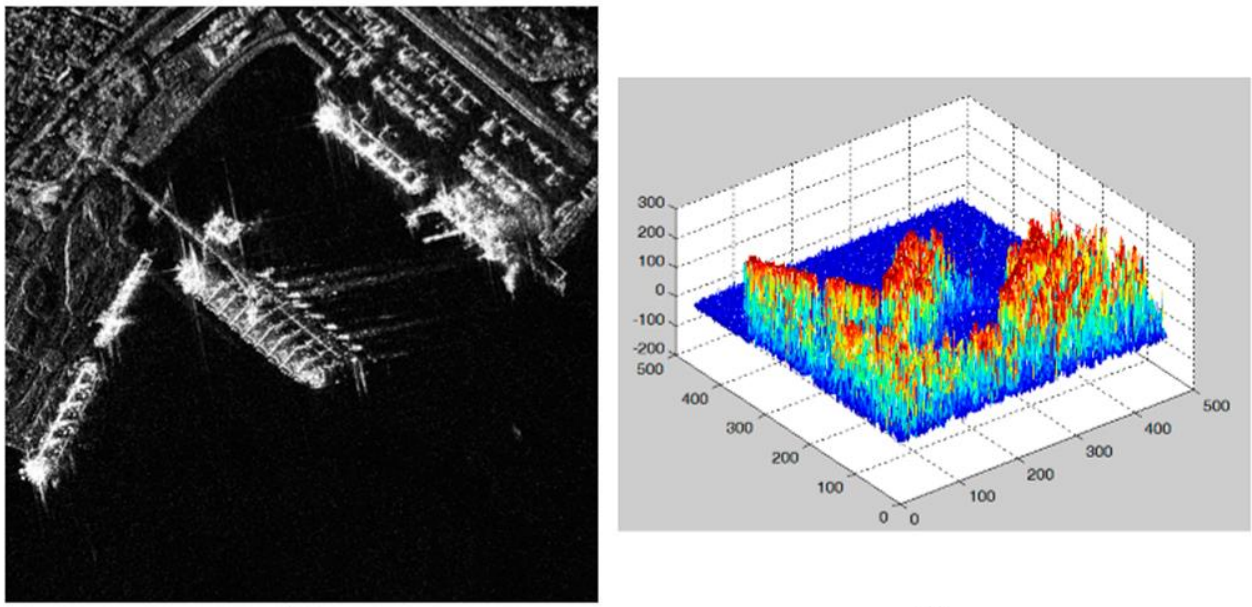
Неформально эту концепцию можно объяснить следующим образом. Представьте все возможные пороговые битовые уровни изображения I. Все пиксели ниже порогового уровня будут ‘белыми’, а выше порогового уровня или равные ему будут ‘черными’. Если бы нам показали анимацию пороговых изображений I_t, с кадром t, соответствующий порогу t, сначала мы увидели бы полностью белое изображение. Впоследствии появлялись бы и растягивались черные пятна, соответствующие локальным максимумам интенсивности. В какой-то момент области соответствующие двум локальным максимумам сольются. И наконец последнее изображение будет полностью черным. Множество всех связанных компонентов всех кадров анимации является множеством всех максимальных областей. Минимальные области могут быть получены инвертированием интенсивности изображения I и запуском того же процесса.



1.1.1. График интенсивности изображения (b), изображение (a) [13]

На рис. 1.1.1 представлен пример использования MSER метода, для извлечения координат судов [13]. На рисунке 1.1.1 (b) изображен график интенсивности изображения (a). Как можно видеть, интенсивность пикселей изображения варьируется от 0 до 255. У чёрных пикселей интенсивность равна 0, и чем светлее области на изображении, тем большему значению функции интенсивности они соответствуют. У белых пикселей значение интенсивности равняется 255.

На рис. 1.1.2 изображена более контрастная фотография (c) и график функции её интенсивности (d). На этом графике можно отчётливо видеть области с большой интенсивностью, соответствующие кораблям и другим объектам на фотографии (c). Потому, что области графика, соответствующие водному пространству, намного менее интенсивные, соответствующие практически чёрному цвету. Однако, стоит заметить, что не смотря на имеющийся шум, на фотографии (a) (рис. 1.1.1), у метода MSER при этом не возникает трудностей, и он корректно определяет максимально стабильные области.



(c) (d)

1.1.2. График интенсивности изображения (c), изображение(d) [13]

Формальное определение понятия MSER и необходимые вспомогательные определения даны ниже.

1.2. Формальное определение MSER

Изображение I является отображением $I : D \subset Z^2 \rightarrow S$. Экстремальные области хорошо определены на изображениях если:

1. S полностью упорядоченное, то есть существует рефлексивное, асимметричное и транзитивное бинарное отношение $\leq$.

В этой работе рассматривается только $S = \{0, 1, \dots, 255\}$, но экстремальные области могут быть определены, к примеру, на вещественных изображениях ($S = R$).

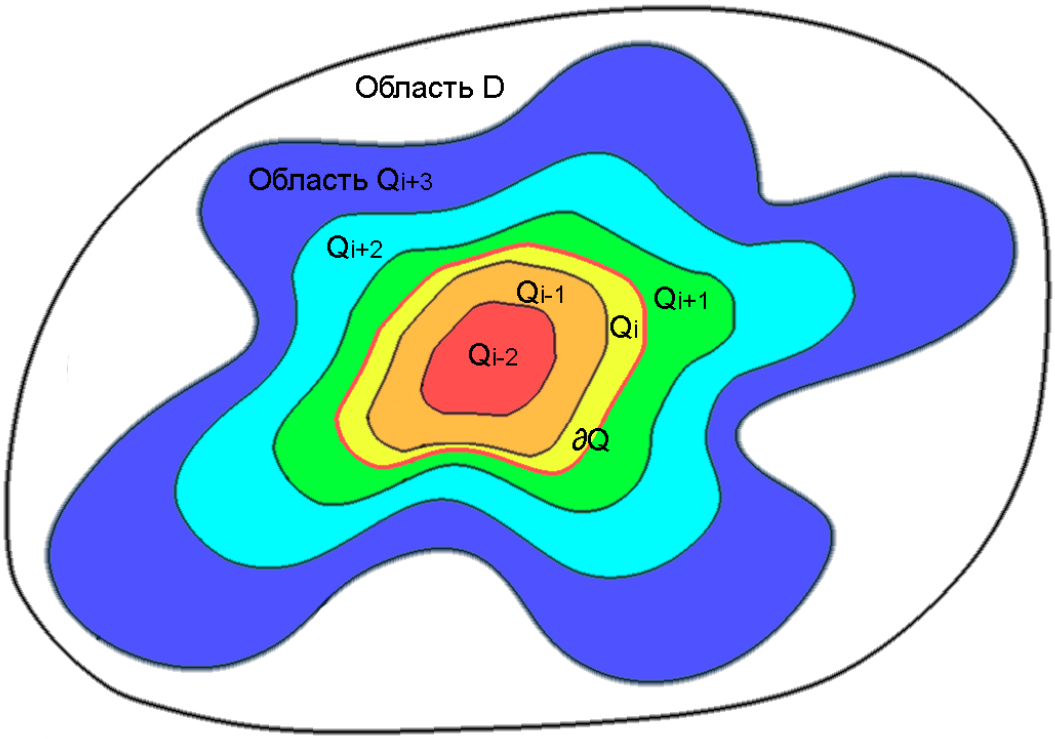
2. Определено отношение смежности (окрестности) $A \subset D \times D$. То есть, любые $p, q \in D$ смежны (pAq) тогда и только тогда, когда $\sum_{i=1}^{d} |p_i - q_i| \leq 1$.

Область Q является непрерывным подмножеством D, то есть для каждого $p, q \in Q$ найдётся последовательность $p, a_1, a_2, \dots, a_n, q$ and $pAa_i, a_iAa_{i+1}, a_nAq$.

(Внешняя) граница области $\partial Q = \{q \in D \setminus Q : \exists p \in Q : qAp\}$, то есть граница ∂Q из области Q является множеством пикселей, смежных по меньшей мере с одним пикселем Q, но не принадлежащим Q.

Экстремальная область $Q \subset D$ это такая область, что для всех $p \in Q$, $q \in \partial Q : I(p) > I(q)$ (область максимальной интенсивности) или $I(p) < I(q)$ (область минимальной интенсивности).

Максимально стабильная экстремальная область (MSER). Пусть $Q_1, \dots, Q_{i-1}, Q_i, \dots$ последовательность вложенных экстремальных областей, то есть $Q_i \subset Q_{i+1}$. Экстремальная область Q_{i^*} является максимально стабильной (устойчивой) тогда и только тогда, когда $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}|/|Q_i|$ имеет локальный минимум в i^* ($|\,.\,|$ означает мощность). $\Delta \in S$.



1.2.1. Максимально стабильные экстремальные области.

На рис. 1.2.2.1 можно увидеть схему нахождения максимально стабильной области. Красным контуром обозначена внешняя граница области ∂Q. Область Q_i является экстремальной областью.

Таким образом, максимально стабильная экстремальная область полностью определяется функцией интенсивности $I(p)$ и внешней границей области ∂Q.

1.3. Свойства MSER

На многих изображениях в определенных регионах локальная бинаризация стабильна в широком диапазоне пороговых значений. Такие области представляют интерес, поскольку они обладают следующими свойствами:

- Инвариантность аффинному преобразованию интенсивностей изображений.
- Ковариантность к сохранению смежности (непрерывности) при преобразовании $T: D \rightarrow D$ в область изображения.
- Стабильность, поскольку выбираются только экстремальные области, которых практически не изменяется в диапазоне пороговых значений.
- Широкомасштабное обнаружение. Так как сглаживания не происходит, обнаруживается как очень маленькая, так и очень большая структура.
- Множество всех экстремальных областей можно перечислить за $O\ (n\ log\ log\ n)$, где n - количество пикселей на изображении.

Перечисление экстремальных регионов происходит следующим образом. Сначала пиксели сортируются по интенсивности. Вычислительная сложность этого шага равна $O\ (n)$. Если диапазон значений S изображения мал, например типичный диапазон $\{0, \ldots, 255\}$, то сортировка может быть реализована как BINSORT [12]. После сортировки, пиксели помещаются в изображение (либо в порядке убывания, либо в порядке возрастания), а список связных компонент и их площадей поддерживается с использованием алгоритма для систем непересекающихся множеств (union-find) [12]. Сложность реализации union-find $O\ (n\ log\ log\ n)$, то есть почти линейная.

В процессе создается структура данных, сохраняющая площадь каждого

связного компонента как функцию интенсивности. Слияние двух компонентов рассматривается как прекращение существования меньшего компонента, и вставка всех пикселей меньшего компонента в более крупный. И наконец, уровни интенсивности, являющиеся локальными минимумами скорости изменения функции площади, выбираются как пороги, создающие максимально устойчивые экстремальные области. Каждая экстремальная область является связной компонентой порогового изображения. На выходе, каждая максимально стабильная экстремальная область представлена расположением локального минимума интенсивности (или максимума) и порога. При обнаружении MSER, мы ищем ряд пороговых значений, которые практически не изменяются по отношению друг к другу, в отличии от остальных пороговых значений. Для каждого множества вложенных экстремальных областей эти пороговые значения могут быть различны. Для некоторых частей изображения может существовать несколько устойчивых порогов. Выбор конкретного из них в оригинальном алгоритме не прописан, поэтому эта задача ложится на самого программиста.

Глава 2. Реализация

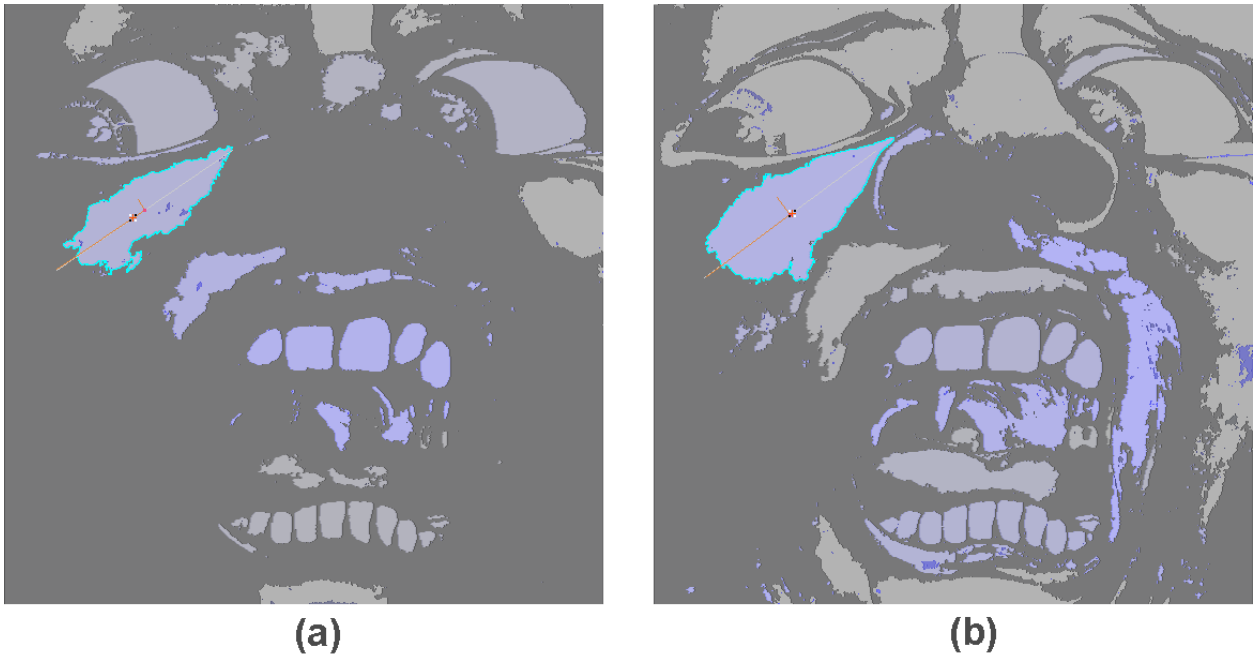
2.1. MSER на Java

В докладе, представленном Иржи Матасом на конференции «British Machine Vision», алгоритм MSER был описан больше как концепция. Поэтому строго стандарта реализации стандартного алгоритма MSER нет. Однако можно принять за некий стандарт библиотеку opencv, так как она является крупнейшей открытой библиотекой и пережила не один десяток версий. Так же на выбор библиотеки повлияло то, что она является самой известной в области компьютерного зрения.

Свою же реализацию MSER, я решил делать на Java, так как этот язык кроссплатформенный и проведение исследований и тестов можно проводить на различных устройствах, работающих под разными операционными системами. Если будет необходимость, можно без проблем портировать программу под Android систему.

В моей программе используются такие объекты как контуры. Так как область или регион (region) представлена ничем иным, как контуром этого региона, то имеет смысл оперировать именно с ними. По ходу выполнения алгоритма постоянно сравниваются контуры двух битовых изображений, располагающихся на соседних уровнях, то есть изображения, получившиеся при бинаризации исходного по соседним порогам. Вообще в алгоритме постоянно используются два экземпляра данных, одни предназначены для первого изображения, другие для второго. Это массивы контуров на каждом изображении, массивы отношений площадей контуров, массивы содержащие контуры с наименьшим отношением площадей, и сами изображения. Для начала сравниваются изображения бинаризованные по 0 и 1 порогу, точнее их

контуры’. После, высчитывается, является ли их отношение максимально стабильным, и в соответствии с этим меняются данные в связанных с изображениями объектах. Далее в объект представляющий первое изображение записывается второе, а в него, в свою очередь, вычисляется и записывается новое изображение на следующем пороге. То же происходит и со связанными с изображениями данными. То есть сравниваются контуры изображений с порогами 0 и 1, 1 и 2, …, 254 и 255. На самом деле сравнивается только один контур, на первом изображении (a), и он же, увеличившийся[1]*, на втором изображении (b) рис. 2.1.1. Далее, когда речь идёт о двух контурах следует понимать, что имеется ввиду одна и та же область (контур) но на разных порогах, такие контуры помечены апострофом «’».

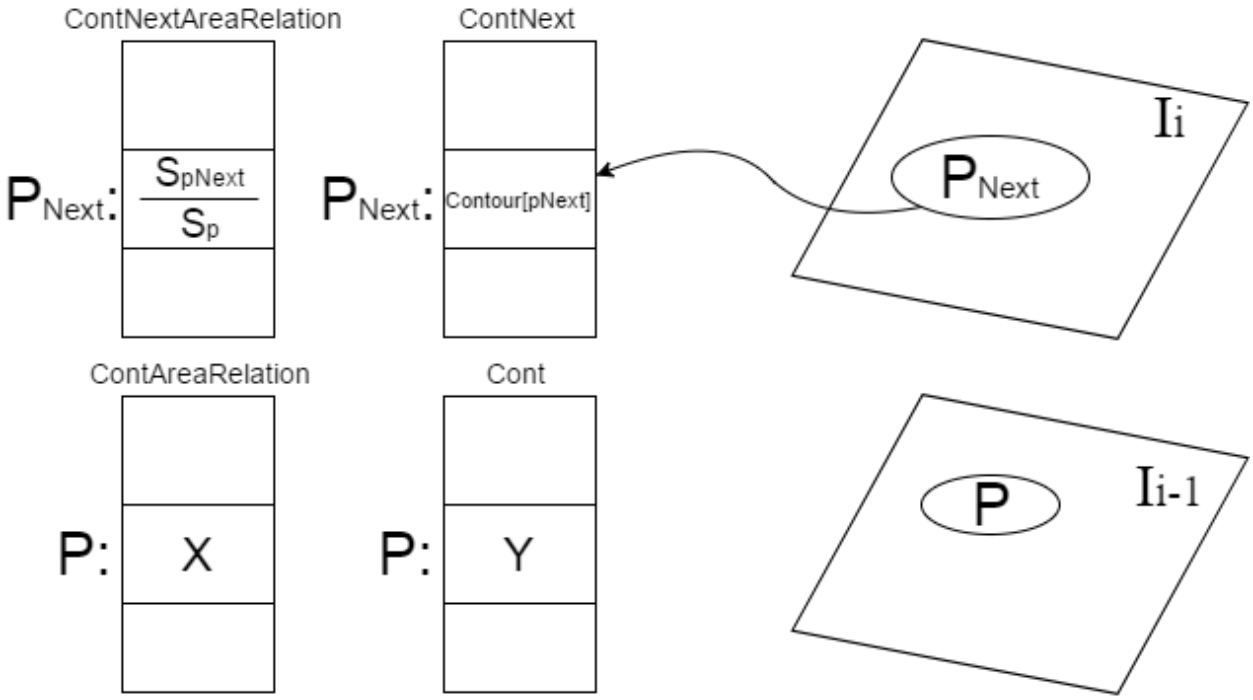


2.1.1. Контур выделен бирюзовым цветом (a), “тот же контур” на более высоком пороге (b).

По ходу сравнения контуров’ наиболее стабильные экстремальные из них, записываются в массив отношений площадей контуров’. Все контуры

[1] Алгоритм сначала проходит по всем чёрным областям. На всех порогах от 0 до 255 чёрные области только растут. Для белых областей алгоритм приходит от 255 порога до 0, в таком случае белые области тоже будут только увеличиваться.

пронумерованы, и индекс массива отношений площадей совпадает с текущем номером контура.

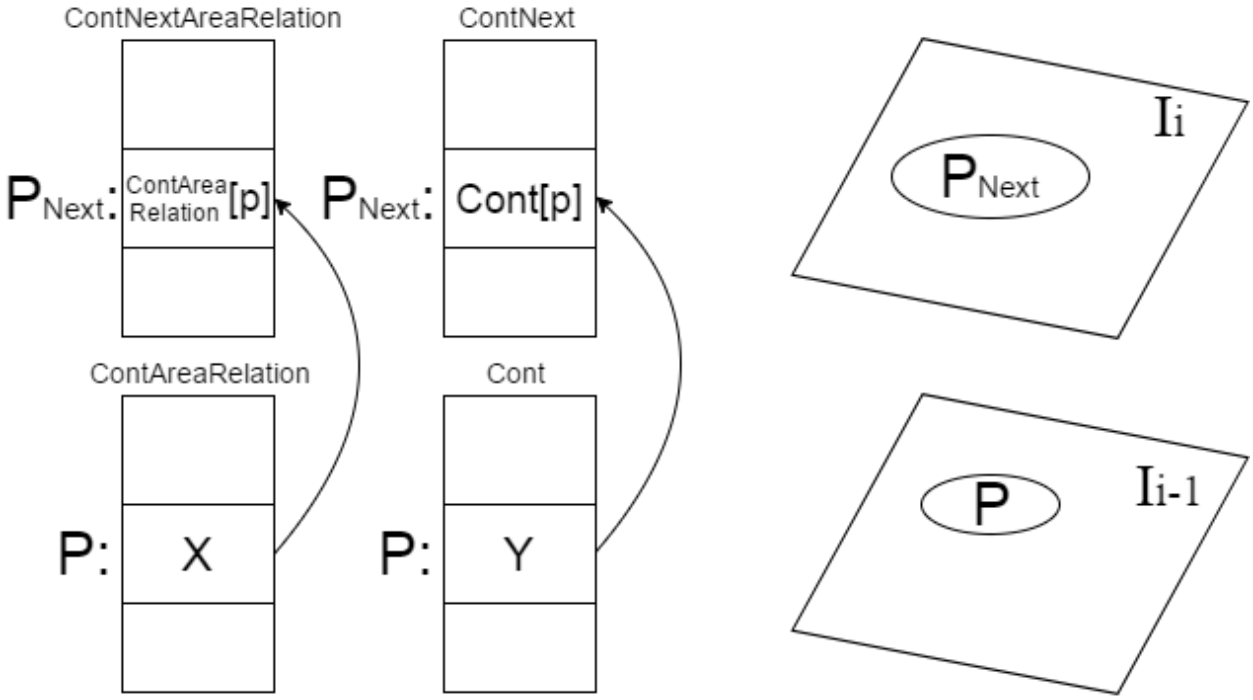


2.1.2. Схема изменения данных при меньшем отношении площадей контуров на i-ом шаге.

На схеме 2.1.2 нарисованы битовые изображения полученные при бинаризации исходного изображения I с порогом I_{i-1} и I_i. На изображении I_{i-1} обозначен контур с номером P, тот же, но увеличенный контур уже с номером P_{Next} находится на изображении I_i. Номера у одного и того же контура на разных изображениях могут отличаться, так как постоянно появляются новые контуры, а старые сливаются или исчезают. Массив $ContAreaRelation$ содержит наименьшее отношение площадей, а в массиве $Cont$ содержатся сами контуры, соответствующие этим отношениям. Так в массиве $ContAreaRelation$ в ячейке с индексом P содержится отношение площадей максимально стабильного региона/контура с этим же контуром, но на предыдущем шаге. Следовательно X может ровняться отношению площадей контуров с номером P изображения I_{i-1} и этого же контура изображения I_{i-2}. Либо X может равняться отношению более стабильной версии контура P с ним

же, но на ещё более ранних этапах.

На каждом шаге ключевыми действиями являются запись новых данных в массивы с отношениями площадей контуров и самими контурами. Если на текущем шаге i, отношение контура с номером P_{Next} изображения I_{i} и контура с номером P изображения $I_{\text{i}-1}$ меньше чем значение, записанное в ячейке с индексом P массива $ContAreaRelation$, то в массив $ContNextAreaRelation$ в ячейку с индексом равному номеру этого контура P_{Next} записывается это отношение. В массив $ContNext$ записывается этот, наиболее стабильный, контур с номер P_{Next} изображения I_{i}.



2.1.3. Схема изменения данных при большем или равном отношении площадей контуров на i-ом шаге.

Если же на текущем шаге i, отношение контура с номером P_{Next} изображения I_{i} и контура с номером P изображения $I_{\text{i}-1}$ больше больше или равно значению, записанному в ячейке с индексом P массива

$ContAreaRelation$, то в массив $ContNextAreaRelation$ записывается значение из ячейки P массива $ContAreaRelation$. Соответственно в массив $ContNext$ записывается значение из массива $Cont$. Схема такой ситуации изображена на рис. 2.1.3.

Таким образом находятся максимально стабильные контуры/регионы. После этого они фильтруются, чтобы исключить слишком маленькие из них, так как они могут являться артефактами изображения.

2.2. MSER на C++

Для написания программы на C++ (C++11) использовалась открытая библиотека opencv версии 3.20 [14].

Программы с реализацией метода MSER с использованием opencv:

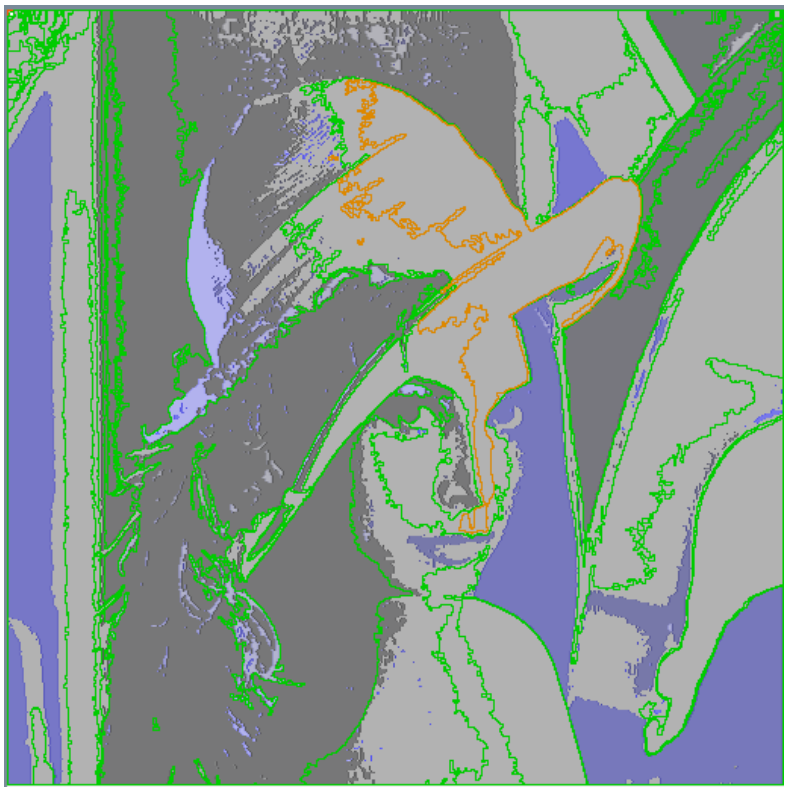
```
int main(int argc, char *argv[])
{
	Mat img = imread(argv[1], 1)

	Ptr<MSER> ms = MSER::create();
	vector<vector<Point> > regions;
	vector<cv::Rect> mser_bbox;
	ms->detectRegions(img, regions, mser_bbox);

	for (int i = 0; i < regions.size(); i++)
	{
		rectangle(img, mser_bbox[i], CV_RGB(0, 255, 0));
	}

	imshow("mser", img);
	return 0;
}
```

Для запуска скомпилированного кода программы, необходимо иметь саму библиотеку opencv, а так же распространяемый пакет Microsoft Visual C++ – Microsoft Visual C++ 2010 Redistributable Package.

Глава 3. Тестирование

3.1. Результаты работы MSER метода на Java.

Метод MSER, реализованный на Java показал очень неплохие результаты. Помимо выполнения своей основной задачи, поиска максимально стабильных экстремальных областей, имеется возможность работы с найденными областями. Благодаря свойствам контуров, таких как отношения главных моментов инерции, отношение квадрата площади к центральному моменту, угол наклона главной оси инерции и сами моменты, их можно использовать в качестве дескриптора для описания области. Так же эти показатели не зависят от масштаба и не требуют никакой обработки.



3.1.1. Результат работы метода MSER на Java.

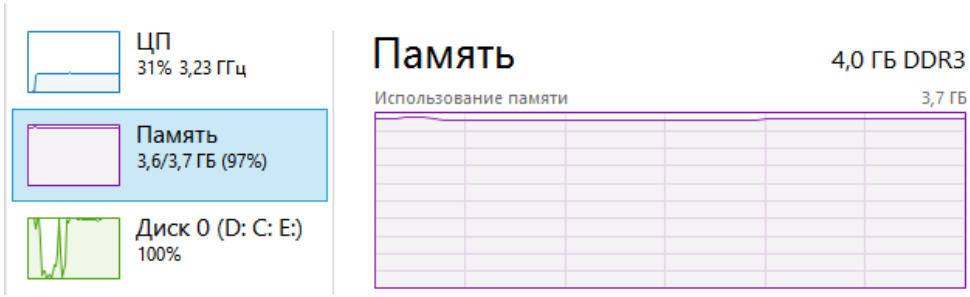
Так же, можно легко оперировать с такими свойствами контура как наибольшее и наименьшее расстояния от края до центра контура, их отношение, наибольшее расстояние между точками контура, длина контура,

отношение наибольшего расстояния и длины, площадь контура, отношение шестнадцати площадей контура к квадрату его длины, координаты центра контура. Результат выполнения программы отображён на рис. 3.1.1.

Сравнение результатов с методом на C++ можно посмотреть в третьем и четвёртом параграфе данной главы.

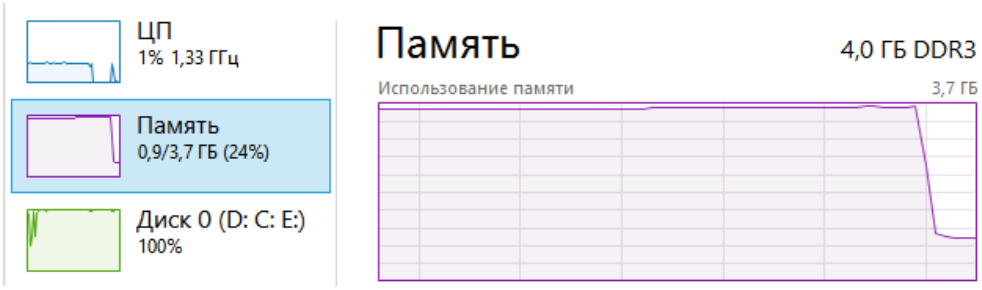
3.2. Результаты работы MSER метода на C++.

С большой вероятностью можно судить о том, что реализация метода MSER в opencv выполнена путём создания 255 изображений с различными порогами интенсивности, так как при обработке больших изображений очень сильно нагружается оперативная память и диск рис. 3.2.1.



3.2.1. Загруженность оперативной памяти и диска, при детектировании.

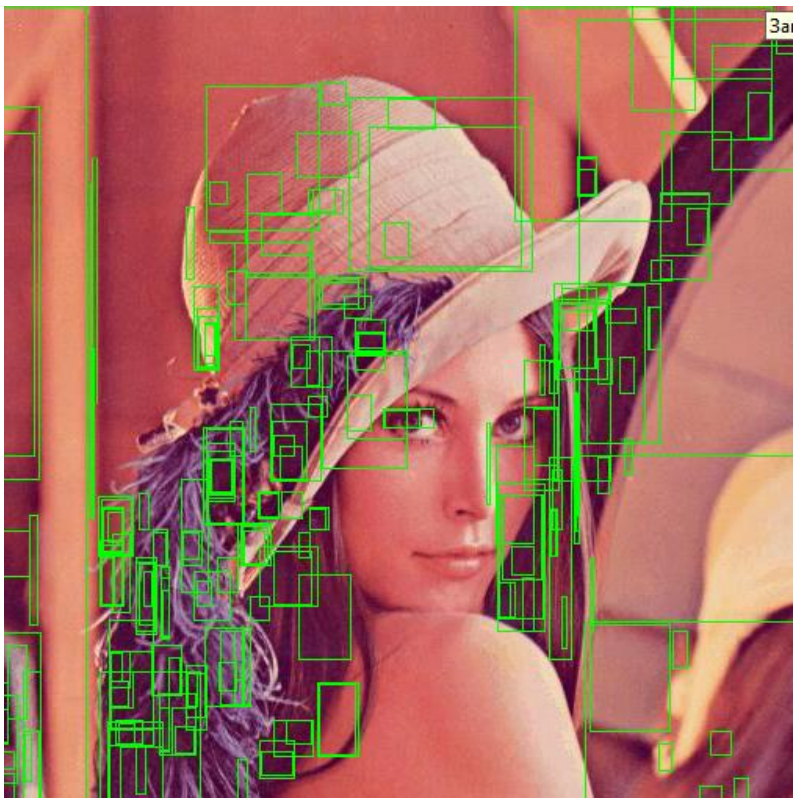
Так, при обработке картинки размерами 4096 на 4096 пикселей, нагрузка на оперативную память возрастала с 0,7 ГБ до максимума 3,7Гб, и незначительно снижаясь после. После выполнения программы нагрузка резко спала рис. 3.2.2.



3.2.2. снижение нагрузки после завершения работы программы.

Реализация программы таким методом может дать значительный прирост к скорости нахождения экстремальных областей, но при больших размерах изображений это может сказаться весьма негативно, в том числе и на времени выполнения программы.

Результат выполнения программы отображён на рис. 3.2.3.

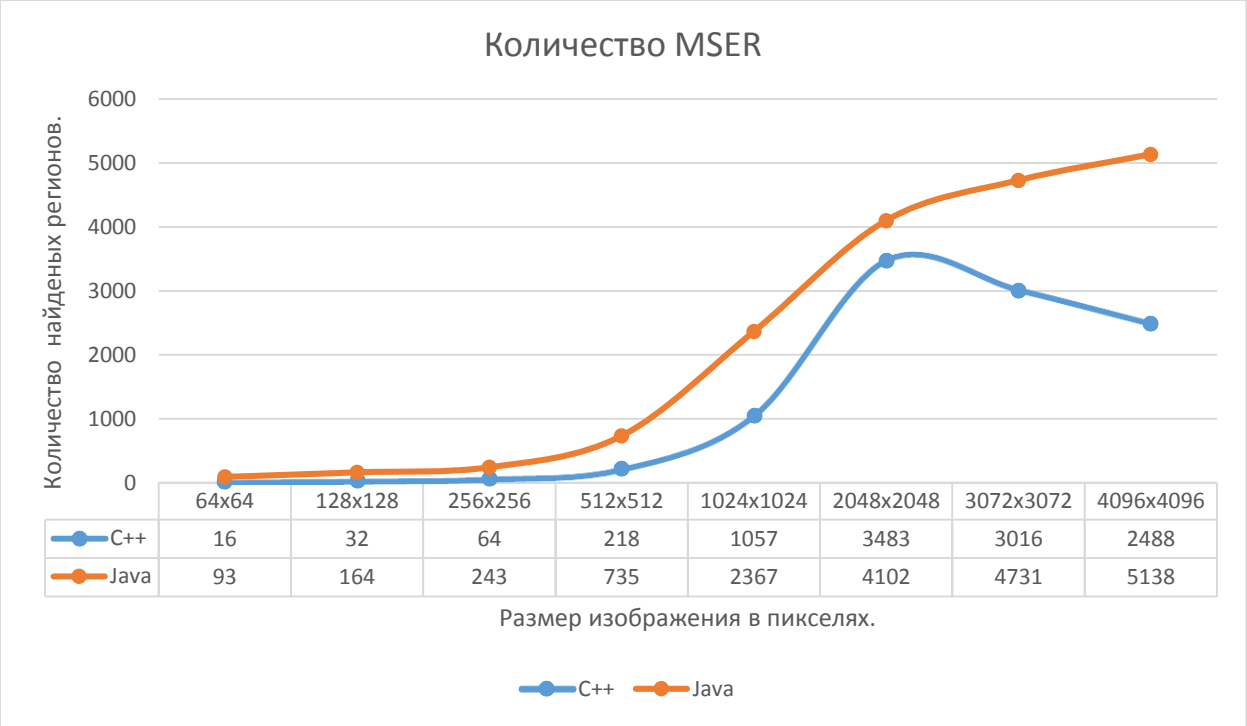


3.2.3. Результат работы метода MSER на C++.

Следует заметить, что для найденных областей, имеется крайне скудный набор свойств: длина, ширина, координаты, в отличии от метода реализованного на Java (все важные свойства контура перечислены в параграфе 1 данной главы).

3.3. Сравнение результатов по количеству найденных регионов.

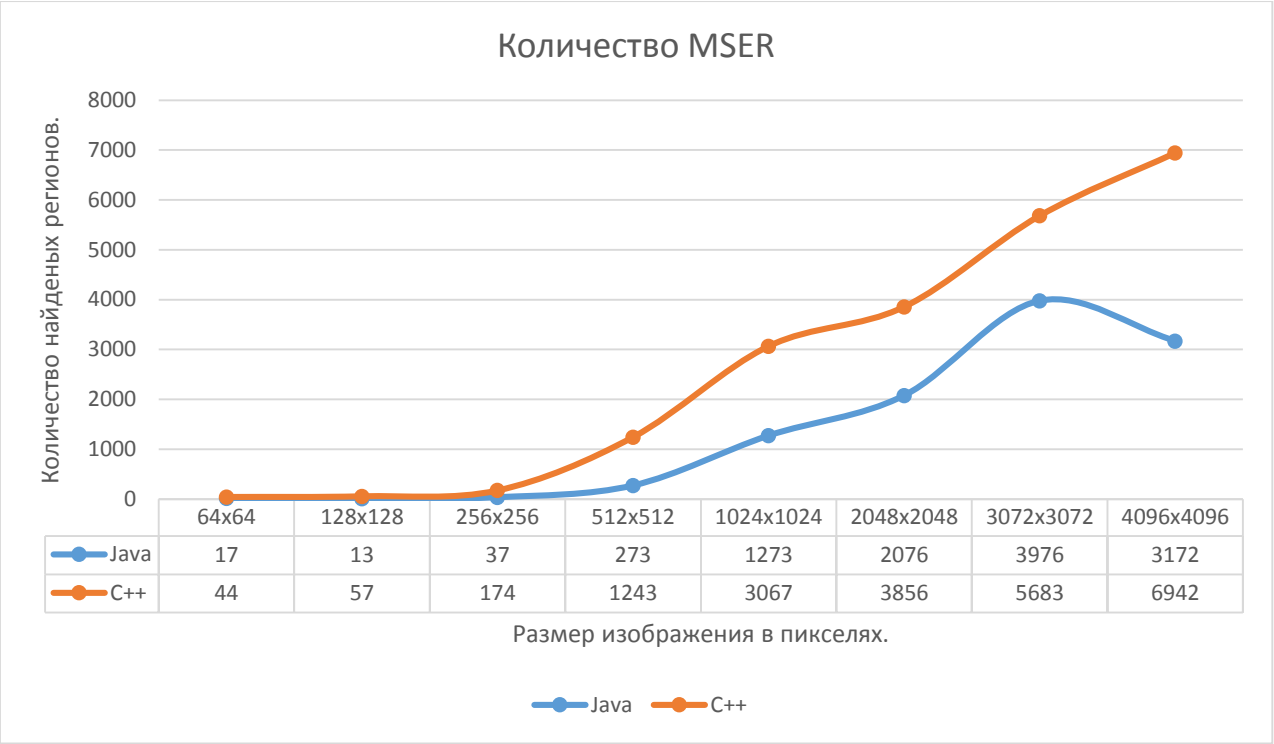
В целом, метод реализованный на C++, с использованием библиотеки opencv работает немного быстрее на маленьких и средних изображениях, но серьёзно замедляется при анализе больших. Скорее всего это связано с недостатком оперативной памяти.



	64x64	128x128	256x256	512x512	1024x1024	2048x2048	3072x3072	4096x4096
C++	16	32	64	218	1057	3483	3016	2488
Java	93	164	243	735	2367	4102	4731	5138

3.3.1. Зависимость количества найденных регионов от размеров одного и того же изображения.

Был проведён ряд тестов, из результатов которых можно сделать вывод, что по количеству найденных областей реализация метода MSER на Java превосходит реализацию на C++. На рис. 3.3.1 изображена таблица с результатами одного из тестов. Так же на графике соответствующему таблице можно увидеть зависимости найденных регионов, от одного и того же изображения, но с разными размерами. На рис. 3.3.2. изображена таблица и график с результатами тестирования по количеству найденных областей на разных изображениях.



3.3.2. Зависимость количества найденных регионов от размеров различных изображений.

3.4. Сравнение результатов по времени работы алгоритмов.

Как уже упоминалось ранее, время работы реализаций метода MSER на Java и C++ не сильно отличаются, однако при больших изображениях время работы программы на C++ сильно возрастает.

	64x64	128x128	256x256	512x5142	1024x1024	2048x2048	3072x3072	4096x4096
Java	0,159	0,864	0,934	3,726	13,826	33,452	194,254	280,143
C++	0,568	0,644	0,849	1,843	6,495	38,217	389,345	406,424

3.4.1. Зависимость времени работы от размеров изображения.

На рис. 3.3.1. изображен график и таблица отображающие результаты одного из тесов. График представляет зависимость времени выполнения программ от размеров изображения.

Выводы

В результате проделанной работы и проведённых исследованиях можно судить о том, что поставленная задача в целом выполнена. Был реализован метод писка максимально стабильных экстремальных регионов, который по определённым параметрам работает лучше, чем метод из библиотеки opencv. Метод MSER реализованный на Java действительно более удобный в плане дальнейшей работы с областями, а так же выигрывает по скорости работы при больших изображениях, и не сильно отличается при обработке маленьких и средних изображений. На графиках так же можно заметить, что при очень маленьких изображениях метод MSER на Java работает быстрей.

Реализация метода на Java позволяет обнаружить весьма большое количество интересующих областей. Если ограничить это множество областей при поиске, то это должно хорошо улучшить скорость выполнения программы.

Продолжая исследования в этой области, можно надеяться, что удастся улучшить алгоритм реализованный на Java, и, возможно, перегнать по скорости работы алгоритм с использованием библиотеки opencv для средних изображений.

Список Литературы

1. Donoser, M. and Bischof, H. Efficient Maximally Stable Extremal Region (MSER) Tracking. // Conference on Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conferenceon, volume 1, pages 553–560. IEEE, 2006.
2. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. // Proceedings of the British Machine Vision Conference, Cardiff, UK, pages 384-396, 2002.
3. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. // International Journal of Computer Vision 65 (2005) pages 43–72
4. Petra Bosilj, Ewa Kijak and Sébastien Lefèvre. Beyond MSER: Maximally Stable Regions using Tree of Shapes. // Proceedings of the British Machine Vision Conference (BMVC), pages 169.1-169.13. BMVA Press, September 2015.
5. Š. Obdržálek and J. Matas. Object recognition using local affine frames on maximally stable extremal regions. // In Toward Category-Level Object Recognition, pages 83–104. Springer, 2006.
6. D. Nistér and H. Stewénius. Scalable Recognition with a Vocabulary Tree. // In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 2161–2168. IEEE, 2006.
7. P.E. Forssén. Maximally stable colour regions for recognition and matching. // In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007.
8. H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B Girod. Robust text detection in natural images with edge-enhanced maximally stable

extremal regions. In Image Processing (ICIP) // 18th IEEE International Conference on, pages 2609– 2612. IEEE, 2011.

9. K. Iqbal, X-C. Yin, X. Yin, H. Ali, and H-W. Hao. Classifier comparison for mserbased text classification in scene images. // In Neural Networks (IJCNN), The 2013 International Joint Conference on, pages 1–6. IEEE, 2013.
10. Shi, C. , Wang, C. , Xiao, B. , Zhang, Y. , Gao, S. Scene text detection using graph model built upon maximally stable extremal regions. // Pattern Recognition Letters. Volume 34, Issue 2, 15 January 2013, Pages 107-116
11. A. Chavez, and D. Gustafson. Color-Based Extensions to MSERs. // ISVC (2), volume 6939 of Lecture Notes in Computer Science, page 358-366. Springer, (2011)
12. R. Sedgewick. Algorithms. // Addison-Wesley, 2nd edition, 1988.
13. Q. Wang, H. Zhu, W. Wu, H. Zhao, N. Yuan. Inshore ship detection using high-resolution synthetic aperture radar images based on maximally stable extremal region. // Journal of Applied Remote Sensing. 9 (1), 095094 (February 24, 2015); doi:10.1117/1.JRS.9.095094
14. Документация по opencv 3.20 // http://docs.opencv.org/3.2.0/